

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

MODUL 3

PEMROGRAMAN JAVA DASAR II



Disusun Oleh:

Ratu Rizki Amelia

105223011

PROGRAM STUDI ILMU KOMPUTER

FAKULTAS SAINS DAN ILMU KOMPUTER

UNIVERSITAS PERTAMINA

2025

I. PENDAHULUAN

Setiap perusahaan memiliki sistem penggajian yang berbeda-beda tergantung pada kebijakan dan struktur kerja yang diterapkan. Dalam sebuah pabrik yang beroperasi dengan sistem kerja tiga shift—pagi, siang, dan malam—perhitungan gaji menjadi lebih kompleks karena adanya perbedaan tarif upah per jam, kemungkinan lembur, serta potongan akibat ketidakhadiran karyawan. Oleh karena itu, diperlukan sebuah sistem yang dapat menghitung gaji karyawan secara akurat dan efisien. Laporan ini membahas pengembangan program berbasis Java yang dirancang untuk membantu proses perhitungan gaji karyawan berdasarkan jumlah jam kerja, shift yang dijalani, serta faktor lain seperti lembur dan absen. Program ini memungkinkan pengguna untuk memasukkan data karyawan, termasuk ID, nama, shift kerja, total jam kerja dalam seminggu, serta jumlah hari absen. Dari data tersebut, sistem akan menghitung gaji akhir dengan mempertimbangkan tarif upah yang berlaku, tambahan upah lembur bagi yang bekerja lebih dari 40 jam, serta potongan gaji bagi karyawan yang bekerja kurang dari 30 jam atau memiliki ketidakhadiran tanpa keterangan. Dengan adanya program ini, diharapkan proses penghitungan gaji dapat dilakukan secara lebih cepat, akurat, dan transparan. Selain itu, sistem ini juga mampu mendeteksi kemungkinan kesalahan dalam pencatatan jam kerja atau jumlah hari absen, sehingga dapat membantu dalam pengelolaan administrasi karyawan secara lebih efektif.

II. VARIABEL

No	Nama Variabel	Tipe Data	Fungsi
1.	id	String	Menyimpan ID karyawan
2.	nama	String	Menyimpan nama karyawan
3.	shift	String	Menyimpan shift kerja karyawan (pagi, siang, malam)
4.	jamKerja	Int	Menyimpan jumlah jam kerja
5.	absen	Int	Menyimpan jumlah hari absen
6.	scanner	Scanner	Mengambil input dari pengguna
7.	jumlahKaryawan	Int	Menyimpan jumlah karyawan yang akan dimasukkan
8.	gajiPerJam	Double	Menyimpan jumlah gaji per jam berdasarkan shift kerja
9.	gajiNormal	Double	Menyimpan gaji dasar berdasarkan jumlah jam kerja
10.	lembur	Double	Menyimpan tambahan gaji jika jam kerja lebih dari 40 jam/minggu
11.	potongan	Double	Menyimpan jumlah potongan gaji jika jam kerja kurang dari 30 jam/minggu
12.	potonganAbsen	Double	Menyimpan jumlah potongan gaji berdasarkan jumlah absen

13.	totalGaji	Double	Menyimpan total gaji akhir setelah perhitungan gaji normal, lembur, dan potongan
-----	-----------	--------	--

III. CONSTRUCTOR DAN METHOD

No	Nama Metode	Jenis Metode	Fungsi
1.	Karyawan()	Constructor	Menginisialisasi objek Karyawan dengan ID, nama, shift, jam kerja, dan absen.
2.	hitungGajiPerJam()	Functional	Menghitung dan mengembalikan gaji per jam berdasarkan shift kerja.
3.	tampilkanGaji()	Procedural	Menghitung total gaji karyawan dan menampilkannya ke layar.
4.	main()	Procedural	Mengelola input data karyawan, menyimpan dalam array, dan menampilkan laporan gaji.

IV. DOKUMENTASI DAN PEMBAHASAN CODE

```
import java.util.Scanner;
```

- Untuk mengambil input dari user

```
class Karyawan {
    String id, nama, shift;
    int jamKerja, absen;
```

- class Karyawan untuk merepresentasikan data karyawan.
- String id, nama, shift untuk menyimpan ID, nama, dan shift kerja karyawan.
- Int jamKerja, absen untuk menyimpan jumlah jam kerja dan jumlah hari absen dalam seminggu.

```
public Karyawan(String idInput, String namaInput, String shiftInput, int jamKerjaInput, int absenInput) {
    id = idInput;
    nama = namaInput;
    shift = shiftInput;
    jamKerja = jamKerjaInput;
    absen = absenInput;
```

- Suatu constructor khusus yang dipanggil saat objek Karyawan dibuat.
- Dan terdapat juga Parameter idInput, namaInput, shiftInput, jamKerjaInput, dan absenInput digunakan untuk mengisi nilai variabel dalam objek.

```
private double hitungGajiPerJam() {
    return switch (shift.toLowerCase()) {
        case "pagi" -> 350000;
        case "siang" -> 400000;
        case "malam" -> 450000;
        default -> 0;
    };
}
```

- Method hitungGajiPerJam() untuk Menghitung gaji per jam berdasarkan shift kerja.
- Menggunakan switch dengan toLowerCase(), Shift akan dicek dalam huruf kecil (pagi, siang, malam).

```
public void tampilkanGaji() {
    double gajiPerJam = hitungGajiPerJam();
    double gajiNormal = jamKerja * gajiPerJam;
    double lembur = (jamKerja > 40) ? (jamKerja - 40) * gajiPerJam * 1.5 : 0;
    double potongan = (jamKerja < 30) ? gajiNormal * 0.1 : 0;
    double potonganAbsen = absen * 100000;
    double totalGaji = gajiNormal + lembur - potongan - potonganAbsen;

    System.out.printf(format:"ID: %s | Nama: %s | Shift: %s | Gaji: Rp%.2f\n", id, nama, shift, totalGaji);
}
```

- Mengambil gaji per jam dengan memanggil hitungGajiPerJam().
- Menghitung gaji normal seperti gajiNormal = jamKerja * gajiPerJam
- Menghitung lembur, Jika lebih dari 40 jam kerja, lembur dihitung sebagai (jamKerja - 40) * gajiPerJam * 1.5
- Menghitung potongan: Jika kurang dari 30 jam kerja, potongan sebesar 10% dari gaji normal.
- Potongan untuk absen dihitung 100000 per hari absen.
- Menghitung total gaji dengan menjumlahkan gaji normal, lembur, lalu dikurangi potongan.

```
System.out.printf(format:"ID: %s | Nama: %s | Shift: %s | Gaji: Rp%.2f\n", id, nama, shift, totalGaji);
```

- Menampilkan hasil gaji karyawan.

```
public class GajiKaryawan {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
    }
}
```

- Deklarasi class GajiKaryawan Digunakan sebagai program utama.
- Method main() suatu Titik awal eksekusi program.
- Membuat objek Scanner untuk mengambil input dari pengguna.

```
System.out.print(s:"Masukkan jumlah karyawan: ");
int jumlahKaryawan = scanner.nextInt();
scanner.nextLine();
```

- Meminta jumlah karyawan yang akan dimasukkan.
- Menggunakan scanner.nextInt() untuk membaca angka.
- Menggunakan scanner.nextLine() untuk membersihkan karakter newline (Enter).

```
for (int i = 0; i < jumlahKaryawan; i++) {
    System.out.printf(format:"\nData Karyawan ke-%d:\n", i + 1);
    System.out.print(s:"ID: ");
    String id = scanner.nextLine();
    System.out.print(s:"Nama: ");
    String nama = scanner.nextLine();
    System.out.print(s:"Shift (pagi/siang/malam): ");
    String shift = scanner.nextLine();
    System.out.print(s:"Jam kerja dalam seminggu: ");
    int jamKerja = scanner.nextInt();
    System.out.print(s:"Hari absen: ");
    int absen = scanner.nextInt();
    scanner.nextLine();
}
```

- Menggunakan perulangan for untuk menginput data karyawan
- Dan untuk membaca ID, nama, shift, jamKerja, dan Jumlah absen

```
if (jamKerja < 0 || jamKerja > 168 || absen < 0 || absen > 7) {
    System.out.println(x:"Data tidak valid. Harap masukkan dataulang.");
    i--;
    continue;
}
```

- Untuk mengecek Validasi, jika data tidak valid maka perulangan for akan mengulang menginput karyawan tsb.

```
System.out.println(x:"\n -- Laporan Gaji Karyawan --");
for (Karyawan karyawan : karyawanList) {
    karyawan.tampilkanGaji();
}
scanner.close();
```

- Menampilkan laporan gaji dari semua karyawan yang sudah dimasukkan.